

Modelling 1

SUMMER TERM 2020



$$A = U D U^T$$

The diagram illustrates the Singular Value Decomposition (SVD) of a matrix A . Matrix A is a 4x4 grid of orange cells. Matrix U is a 4x4 grid of blue cells. Matrix D is a 4x4 grid of light pink cells with diagonal elements $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ in red. Matrix U^T is a 4x4 grid of blue cells. An equals sign is placed between A and U .

LECTURE 9

Eigen- and Singular Values

Eigenvectors and Eigenvalues

Eigenvectors & Eigenvalues

Definition:

- Linear map \mathbf{A} , non-zero vector \mathbf{x} with

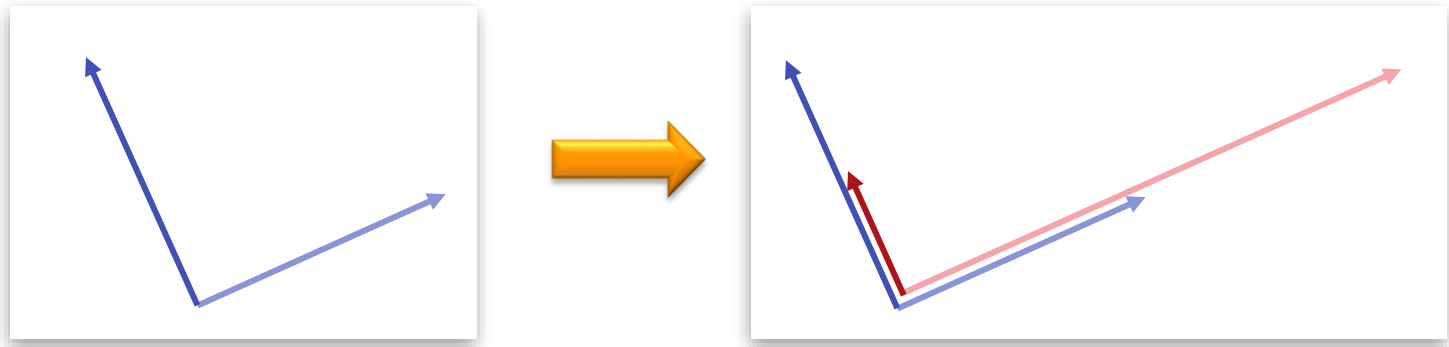
$$\mathbf{Ax} = \lambda\mathbf{x}$$

- λ is an *eigenvalue* of \mathbf{A}
- \mathbf{x} is the corresponding *eigenvector*

Example

Intuition:

- In the direction of an eigenvector, the linear map acts like a scaling



- Example:
 - Two eigenvalues (0.5 and 2)
 - Two eigenvectors
- Standard basis $\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$: *not eigenvectors*

Eigenvectors & Eigenvalues

Theorem

- All real, *symmetric* matrices can be diagonalized
 - Orthogonal eigenbasis $\mathbf{U} = (\mathbf{u}_1 | \dots | \mathbf{u}_d)$
 - $\mathbf{A} = \mathbf{U} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{pmatrix} \mathbf{U}^T$
- Symmetric matrices encode only non-uniform scaling

Diagonalization

Eigenvalue decomposition (diagonalization)

$$\mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{U}^T$$

The diagram shows the decomposition of a symmetric matrix \mathbf{A} into an orthogonal matrix \mathbf{U} , a diagonal matrix \mathbf{D} , and its transpose \mathbf{U}^T . The matrix \mathbf{A} is represented by a 4x4 grid of orange cells and is labeled "symmetric". The matrix \mathbf{U} is represented by a 4x4 grid of blue cells and is labeled "orthogonal". The matrix \mathbf{D} is represented by a 4x4 grid of red cells with diagonal elements $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ and zero off-diagonal elements. The matrix \mathbf{U}^T is represented by a 4x4 grid of blue cells and is labeled "orthogonal".

Always possible for symmetric matrices

- Symmetric: $\mathbf{A}^T = \mathbf{A}$

Computation

Simple algorithm

- “Power iteration” for symmetric matrices
- Computes largest eigenvalue even for large matrices
- Algorithm:
 - Start with a random vector (maybe multiple tries)
 - Repeatedly multiply with matrix
 - Normalize vector after each step
 - Repeat until ratio before / after normalization converges (this is the eigenvalue)
- Intuition:
 - Largest eigenvalue = “dominant” component/direction

Powers of Matrices

What happens:

- A symmetric matrix can be written as:

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^T = \mathbf{U} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \mathbf{U}^T$$

- Taking it to the k -th power yields:

$$\mathbf{A}^k = \mathbf{U}\mathbf{D}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{U}^T \dots \mathbf{U}\mathbf{D}\mathbf{U}^T = \mathbf{U} \begin{pmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k \end{pmatrix} \mathbf{U}^T$$

- EV's key to understanding powers of matrices

Generalization: SVD

Singular value decomposition:

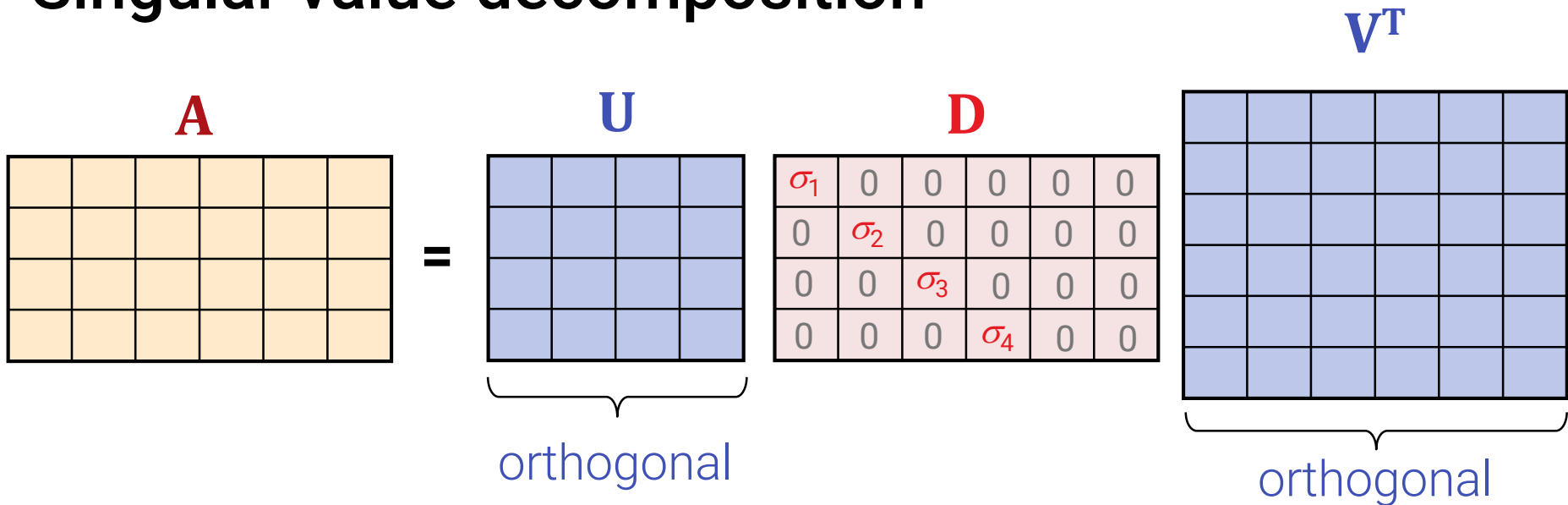
- For any real matrix **A**

$$\mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{V}^T$$

- **U**, **V** are orthogonal
- **D** is a diagonal
- Diagonal entries σ_i : “*singular values*”
- **U** and **V** are different in general
 - For symmetric matrices, they are the same
 - Then: singular values = eigenvalues
- Analogous for linear operators (∞ -dim)

Singular Value Decomposition

Singular value decomposition



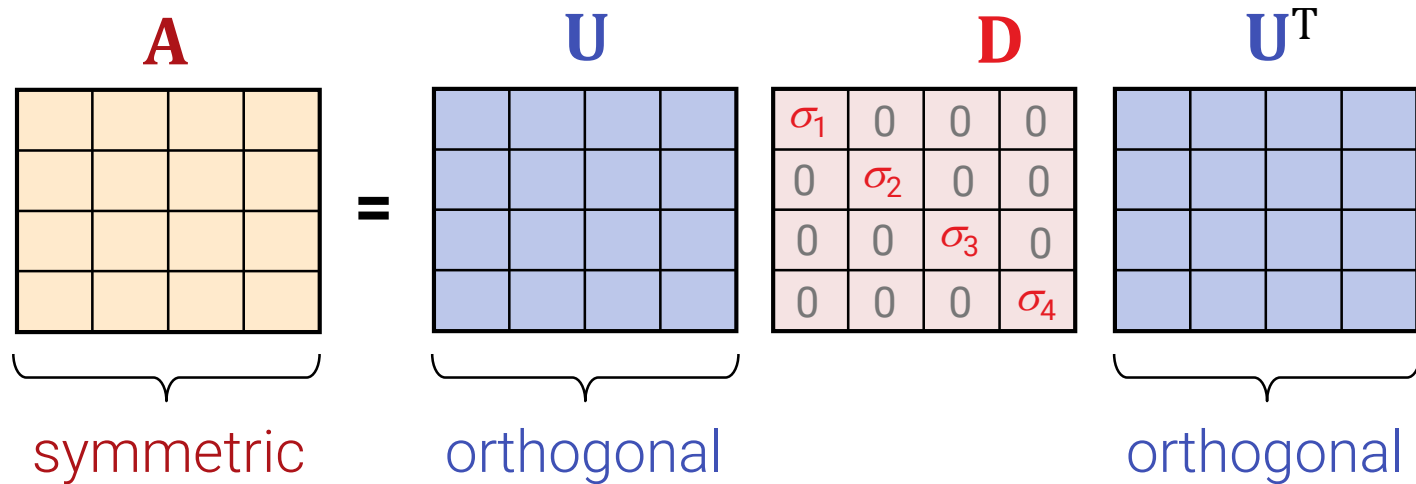
*"the Swiss army knife
of linear algebra"*



[wikipedia user Bisco]

Comparison: Diagonalization

Eigenvalue decomposition (diagonalization)



(For symmetric matrices)

Singular Value Decomposition

SVD Solver

- For full rank, square **A**:

$$\mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{V}^T$$

$$\Rightarrow \mathbf{A}^{-1} = (\mathbf{U} \mathbf{D} \mathbf{V}^T)^{-1} = (\mathbf{V}^T)^{-1} \mathbf{D}^{-1} (\mathbf{U}^{-1}) = \mathbf{V} \mathbf{D}^{-1} \mathbf{U}^T$$

- Numerically very stable
- More expensive than iterative solvers
- General **A** possible (least-squares / pseudo-inverse)
 - More later